AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method for bounding the solution set of a system of linear equations **Ax** = **b**, wherein **A** is an interval matrix and **b** is an interval vector, the method comprising:

receiving the system of linear equations **Ax** = **b**;

storing **Ax** = **b** in a memory in a computer system;

preconditioning the set of linear equations **Ax** = **b** by multiplying ~~through~~ both side of the linear equations by a matrix **B** to produce a preconditioned set of linear equations **BAx=Bb**, wherein the set of linear equations is a representation of a global optimization problem;

substituting $\mathbf{M}_0$ = **BA** and **r** = **Bb** to produce $\mathbf{M}_0\mathbf{x}$ = **r**;

widening the matrix $\mathbf{M}_0$ to produce a widened matrix **M**, wherein the midpoints of the interval elements of **M** form the identity matrix; and

using **M** and **r** to compute ~~the hull~~ a hull **h** of the system **Mx** = **r**, which bounds the solution set of the system $\mathbf{M}_0\mathbf{x}$ = **r**;

wherein the interval operations involved in bounding the solution set are performed using a special-purpose interval arithmetic unit configured to perform interval arithmetic operations.

2. (Previously presented) The method of claim 1, wherein the method further comprises computing the matrix **B** by:

computing an approximate center $\mathbf{A_C}$ of the interval elements of matrix $\mathbf{A}$; and

forming $\mathbf{B}$ by computing an approximate inverse of $\mathbf{A_C}$, $\mathbf{B} = (\mathbf{A_C})^{-1}$.

3 (Canceled).

4. (Previously presented) The method of claim 1, further comprising assuring that $sup(r_i) \geq 0$ by changing the sign of r_i [[(]]and x_i[[)]] if necessary, wherein r_i is an element of $\mathbf{r}$.

5. (Original) The method of claim 1, further comprising:

determining if $\mathbf{M}$ is regular; and

using the Gauss-Seidel process for computing the hull $\mathbf{h}$ if $\mathbf{M}$ is not regular.

6. (Currently amended d) A computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for bounding the solution set of a system of linear equations $\mathbf{Ax} = \mathbf{b}$, wherein $\mathbf{A}$ is an interval matrix and $\mathbf{b}$ is an interval vector, the method comprising:

receiving the system of linear equations $\mathbf{Ax} = \mathbf{b}$;

storing $\mathbf{Ax} = \mathbf{b}$ in a memory in a computer system;

preconditioning the set of linear equations $\mathbf{Ax} = \mathbf{b}$ by multiplying ~~through~~ both side of the linear equations by a matrix $\mathbf{B}$ to produce a preconditioned set of linear equations $\mathbf{BAx=Bb}$, wherein the set of linear equations is a representation of a global optimization problem;

substituting $\mathbf{M_0} = \mathbf{BA}$ and $\mathbf{r} = \mathbf{Bb}$ to produce $\mathbf{M_0x} = \mathbf{r}$;

widening the matrix $\mathbf{M}_0$ to produce a widened matrix **M**, wherein the midpoints of the interval elements of **M** form the identity matrix; and

using **M** and **r** to compute ~~the hull~~ a hull **h** of the system $\mathbf{Mx} = \mathbf{r}$, which bounds the solution set of the system $\mathbf{M}_0\mathbf{x} = \mathbf{r}$;

wherein the interval operations involved in bounding the solution set are performed using a special-purpose interval arithmetic unit configured to perform interval arithmetic operations.

7. (Previously presented) The computer-readable storage medium of claim 6, wherein the method further comprises computing the matrix **B** by:

computing an approximate center $\mathbf{A}_C$ of the interval elements of matrix **A**; and

forming **B** by computing an approximate inverse of $\mathbf{A}_C$, $\mathbf{B} = (\mathbf{A}_C)^{-1}$.

8. (Previously presented) The computer-readable storage medium of claim 6, wherein using **M** and **r** to compute the hull **h** involves:

forming **P** as an inverse of the left endpoint of **M**;

forming $c_i = 1/(2P_{ii} - 1)$ for $i = 1, \ldots, n$;

forming $z_i = (inf(r_i)+sup(r_i))P_{ii} - e_i^T\mathbf{P}sup(\mathbf{r})$, wherein e_i^T is a unit vector in which the i-th element is 1 and other elements are 0, and wherein r_i is an element of **r**;

setting $inf(h_i) = c_i z_i$ if $z_i > 0$;

setting $inf(h_i) = z_i$ if $z_i \le 0$; and

setting $sup(\mathbf{h}) = \mathbf{P}sup(\mathbf{r})$.

9. (Previously presented) The computer-readable storage medium of claim 6, wherein the method further comprises assuring that $sup(r_i) \ge 0$ by changing the sign of r_i [[(]]and x_i[[)]] if necessary, wherein r_i is an element of **r**.

10. (Original) The computer-readable storage medium of claim 6, wherein the method further comprises:

determining if $\mathbf{M}$ is regular; and

using the Gauss-Seidel process for computing the hull $\mathbf{h}$ if $\mathbf{M}$ is not regular.

11. (Currently amended) An apparatus that bounds the solution set of a system of linear equations $\mathbf{Ax} = \mathbf{b}$, wherein $\mathbf{A}$ is an interval matrix and $\mathbf{b}$ is an interval vector, comprising:

a receiving mechanism configured to receive the system of linear equations $\mathbf{Ax} = \mathbf{b}$;

a special purpose arithmetic unit configured to perform interval computations;

a storing mechanism configured to store $\mathbf{Ax} = \mathbf{b}$ in a memory in a computer system;

a preconditioning mechanism within the special purpose arithmetic unit that is configured to precondition the set of linear equations $\mathbf{Ax} = \mathbf{b}$ by multiplying ~~through~~ both side of the linear equations by a matrix $\mathbf{B}$ to produce a preconditioned set of linear equations $\mathbf{BAx}=\mathbf{Bb}$, wherein the set of linear equations is a representation of a global optimization problem;

a substituting mechanism within the special purpose arithmetic unit that is configured to substitute $\mathbf{M}_0 = \mathbf{BA}$ and $\mathbf{r} = \mathbf{Bb}$ to produce $\mathbf{M}_0\mathbf{x} = \mathbf{r}$;

a widening mechanism within the special purpose arithmetic unit that is configured to widen the matrix $\mathbf{M}_0$ to produce a widened matrix $\mathbf{M}$, wherein the midpoints of the interval elements of $\mathbf{M}$ form the identity matrix; and

a hull computing mechanism within the special purpose arithmetic unit that is configured to use $\mathbf{M}$ and $\mathbf{r}$ to compute ~~the hull~~ a hull $\mathbf{h}$ of the system $\mathbf{Mx} = \mathbf{r}$, which bounds the solution set of the system $\mathbf{M}_0\mathbf{x} = \mathbf{r}$;

wherein the interval operations involved in bounding the solution set are performed using the special-purpose interval arithmetic unit.

12. (Previously presented) The apparatus of claim 11, wherein the preconditioning mechanism is configured to:

compute an approximate center $\mathbf{A_C}$ of the interval elements of matrix $\mathbf{A}$; and to

form $\mathbf{B}$ by computing an approximate inverse of $\mathbf{A_C}$, $\mathbf{B} = (\mathbf{A_C})^{-1}$.

13. (Previously presented) The apparatus of claim 11, wherein the hull computing mechanism is configured to:

form $\mathbf{P}$ as an inverse of the left endpoint of $\mathbf{M}$;

form $c_i = 1/(2P_{ii} - 1)$ for $i = 1, \ldots, n$;

form $z_i = (inf(r_i)+sup(r_i))P_{ii} - e_i^T\mathbf{P}sup(\mathbf{r})$, wherein e_i^T is a unit vector in which the i-th element is 1 and other elements are 0, and wherein r_i is an element of $\mathbf{r}$;

set $inf(h_i) = c_i z_i$ if $z_i > 0$;

set $inf(h_i) = z_i$ if $z_i \leq 0$; and to

set $sup(\mathbf{h}) = \mathbf{P}sup(\mathbf{r})$.

14. (Previously presented) The apparatus of claim 11, wherein the preconditioning mechanism is configured to assure that $sup(r_i) \geq 0$ by changing the sign of r_i [[(]]and x_i[[)]] if necessary, wherein r_i is an element of $\mathbf{r}$.

15. (Original) The apparatus of claim 11, wherein the preconditioning mechanism is configured to:

determine if $\mathbf{M}$ is regular; and to

terminate the process of computing the hull $\mathbf{h}$ if $\mathbf{M}$ is not regular.

16. (Currently amended) A method for bounding the solution set of a system of linear equations **Ax** = **b** by multiplying ~~through~~ both side of the linear equations by the matrix **B** to produce a preconditioned set of linear equations **BAx=Bb**, wherein the set of linear equations is a representation of a global optimization problem, the method comprising:

receiving the system of linear equations **Ax** = **b**;

storing **Ax** = **b** in a memory in a computer system;

substituting $\mathbf{M}_0 = \mathbf{BA}$ and $\mathbf{r} = \mathbf{Bb}$ producing $\mathbf{M}_0\mathbf{x} = \mathbf{r}$;

assuring that $sup(r_i) \geq 0$ by changing the sign of r_i (and x_i) if necessary;

widening the matrix $\mathbf{M}_0$ to produce a widened matrix **M**, wherein the midpoints of the interval elements of **M** form the identity matrix; and

using **M** and **r** to compute ~~the hull~~ a hull **h** of the system **Mx** = **r**, which bounds the solution set of the system $\mathbf{M}_0\mathbf{x} = \mathbf{r}$ by,

forming **P** as an inverse of the left endpoint of **M**,

forming $c_i = 1/(2P_{ii} - 1)$ for $i = 1, \ldots, n$,

forming $z_i = (inf(r_i)+sup(r_i))P_{ii} - e_i^T\mathbf{P}sup(\mathbf{r})$,

wherein e_i^T is a unit vector in which the i-th element is 1 and other elements are 0, and wherein r_i is an element of **r**,

setting $inf(h_i) = c_i z_i$ if $z_i > 0$,

setting $inf(h_i) = z_i$ if $z_i \leq 0$, and

setting $sup(\mathbf{h}) = \mathbf{P}sup(\mathbf{r})$;

wherein the interval operations involved in bounding the solution set are performed using a special-purpose interval arithmetic unit configured to perform interval arithmetic operations.

17. (Original) The method of claim 16, further comprising:

determining if **M** is regular; and

using the Gauss-Seidel process for computing the hull **h** if **M** is not regular.

18. (Previously presented) The method of claim 16, wherein the method further comprises computing the matrix **B** by:

computing an approximate center $\mathbf{A}_C$ of the interval elements of matrix **A**; and

forming **B** by computing an approximate inverse of $\mathbf{A}_C$, $\mathbf{B} = (\mathbf{A}_C)^{-1}$.